Securing Web Servers*

Peter Mell, David Ferraiolo
National Institute of Standards and Technology
Computer Security Division
September, 1999

A Web site is a powerful tool that enables businesses, government, and private users to share information and conduct business on the Internet. Organizations – small and large, private and public – are devoting many resources to creating attractive, attention-getting Web sites, but they may be neglecting basic security controls. Recent attacks on Web sites have shown that the computers that support Web sites are vulnerable to attacks that can range from minor nuisances to significant interruptions of service. This *ITL Bulletin* discusses the most commonly employed methods for protecting Web servers and provides practical guidance on steps that organizations can take to reduce the threat of attacks.

Creating a Plan to Secure Your Web Server

While most incidents cause minor embarrassment or inconvenience, it is possible for an intruder to cause real problems and severe losses. Every organization should establish a security program that assesses the risks of attacks and takes steps to reduce the risks to an acceptable level. Each organization has to decide its sensitivity to risk and how open it wants to be to the external world. When resources are limited, the cost of security incidents should be considered, and the investment in protective measures should be concentrated on areas of highest sensitivity.

There are three levels of Web security techniques that can be applied:

Level 1: Minimum Security
1. Upgrading Software/Installing Patches
2. Using Single Purpose Servers
3. Removing Unnecessary Applications

Level 2: Penetration Resistance
1. External Firewalls
2. Remote Administration Security
3. Restrict Server Scripts
4. Web Server Shields with Packet Filtering
5. Education and Personnel Resource Allocation
6. Techniques listed in level 1

Level 3: Attack Detection and Mitigation
1. Separation of Privilege
2. Hardware-Based Solutions

* Published in the National Institute of Standards and Technology's Information Technology Bulletin, September 1999.

3. Internal Firewalls
4. Network-Based Intrusion Detection
5. Host-Based Intrusion Detection
6. Techniques listed in level 2

Techniques to Secure Web Servers

The most common methods for protecting Web servers include:

- Removal of unnecessary software,
- Detection of attacks upon a Web server,
- Correction of flaws in remaining software,
- Restriction of an attacker's actions once a part of a Web server is compromised, and
- Protection of the rest of the network if a Web server is compromised.

Upgrading Software/Installing Patches

One of the simplest and yet most effective techniques for reducing risk is the installation of the latest software updates and patches. Web servers should be frequently (sometimes daily) examined to determine what software needs to be updated or patched. (NIST is actively working with other government agencies to develop tools to assist in the finding and applying of patches. When available, details will appear on the NIST Computer Security Resource Clearinghouse [http://csrc.nist.gov].) Any software on a Web server that an attacker could use to penetrate the system must be regularly updated. Software in this category includes the operating system, servers or any software that receives network packets, software running as root or administrator, and security software. The following process should be followed:

- Make a list of such software and write down the associated version numbers.
- Find the Web page for each piece of software and make sure that you have installed the latest version.
- Find and install the available patches for the applicable version of the software. Each software vendor provides unique instructions on how to install its patches and usually these instructions are very simple. Be careful to follow vendor instructions; patches must often be installed in a set sequence for the process to work.
- Verify that patched software functions correctly.

Using Single-Purpose Servers

Organizations should run Web servers on computers dedicated exclusively to that task. A common mistake is to try to save money by running multiple servers on the same host. For example, it is not uncommon to run an e-mail server, Web server, and database server on the same computer. However, each server run on a host provides an attacker with avenues for attack. Each newly installed server then increases the organization's reliance upon that host while simultaneously decreasing its security. Given the decreasing cost of hardware and the increasing importance of having fast Web servers, it is generally effective to buy a dedicated host for each Web server. Also, in situations where a Web server constantly interacts with a database, it is best to use two separate hosts.

Removing Unnecessary Applications

All privileged software not specifically required by the Web server should be removed. For the purposes of this document, privileged software is defined as software that runs with administrator privileges or that receives packets from the network. Operating systems often run a variety of privileged programs by default. Many systems administrators are not even aware of the existence of many of these programs. Each privileged program provides another avenue by which an attacker can compromise a Web server. It is therefore crucial that Web servers be purged of unnecessary programs. For greater security and because it is often difficult to identify what software is privileged, many systems administrators remove all software not needed by a Web server.

External Firewalls

Install public Web servers outside of an organization's firewall. In this configuration, the firewall prevents the Web server from sending packets into an organization's network. If an attacker on the Internet penetrates the external Web server, they have no more access to the organization's internal network than they had before. If a Web server is inside the organization's firewall and is penetrated by an attacker on the Internet, the attacker can use the Web server as a launching point for attacks on the internal systems. Thus, these attacks would completely bypass the security provided by the firewall.

Remote Administration Security

Since it is often inconvenient to administer a host from the physical console, system administrators often install software on Web servers to allow remote administration. From a security perspective, this practice is dangerous and should be minimized or eliminated. In order to increase the security where this practice is necessary:

- Encrypt remote administration traffic such that attackers monitoring network traffic cannot obtain passwords or inject malicious commands into conversations.
- Use packet filtering (see description below) to allow remote administration only from a designated set of hosts.
- Maintain this designated set of hosts at a higher degree of security than normal hosts.
- Do not use packet filtering as a replacement for encryption since attackers can spoof Internet Protocol (IP) addresses. (With IP spoofing, an attacker lies about their location by sending messages from an IP address other than their own.)

Restrict Server Scripts

Most Web sites contain scripts (small programs) created locally by Web site developers. A Web server runs these scripts when a user requests a particular page. Attackers can use these scripts to penetrate Web sites by finding and exercising flaws in the code. To find such flaws, an attacker does not necessarily need the script source code. Scripts must be carefully written with security in mind and system administrators should inspect them before placing them on a Web site. Do not allow scripts to run arbitrary commands on a system or to launch insecure (or non-patched) programs. Scripts should restrain users to doing a small set of well-defined tasks. They should carefully restrict the size of input parameters so that an attacker cannot give a script more data than it expects. If an attacker is allowed to do this, a system can often be penetrated using a technique called buffer overflow. (With a buffer overflow attack, an attacker convinces a Web server to run arbitrary code by giving it more information than it expected to receive.) Run scripts

with non-administrator privileges to prevent an attacker from compromising the entire Web server in the event that a script contains flaws.

Web Server Shields with Packet Filtering

A router set up to separate a Web server from the rest of the network can shield a Web server from many attacks. The router can thwart attacks before they reach the Web server by dropping all packets that do not access valid Web server services. Typically, the router should drop all network packets that do not go either to the Web server (port 80) or to the remote administration server being used. For additional security, only allow a pre-approved list of hosts to send traffic to a Web server's remote administration server. By doing so, an attacker can only compromise a Web server using the remote administration server via a restricted set of network paths. The filtering router shield offers similar protection to that of removing all unneeded software from a host since it prevents an attacker from requesting certain vulnerable services. Be aware that setting up a router with many filtering rules may noticeably slow its ability to forward packets.

Education and Personnel Resource Allocation

Attackers are able to penetrate most Web servers because the systems administrators are either not knowledgeable about Web server security or did not take the time to properly secure the system. Web site administrators must be trained about Web server security techniques and rewarded for spending time securing the site. Several excellent books and training seminars exist to aid administrators in securing Web sites.

Separation of Privilege

Regardless of the security measures established for a Web server, penetration may still occur. If this happens, it is important to limit the attacker's actions on the penetrated host. Separation of privilege is a key concept for restricting actions once a part of the host is penetrated. To establish such control, partition the various host resources among a set of user accounts. An attacker who penetrates some software will then be limited to acting within that single user account instead of having control over the entire system. For example, a Web server can run as one user, but the Web pages can be owned by another user and with the Web server given read-only access. Then, if attackers penetrate the Web server, they cannot change the Web pages owned by other users. Likewise, intrusion detection software can run as another user to protect it from being modified by an attacker penetrating the Web server user. For the best security, run the Web server process as a user that has write privilege only in a few privately owned temporary directories. This requires storing the Web server software as read-only under one user but running it as a different user.

Hardware-Based Solutions

Hardware can implement separation of privilege concepts with a greater degree of security than software because hardware is not as easily modified as software. With software implementations, if the underlying operating system is penetrated, the attacker has complete control of all files on a Web server. Using read-only external hard disks or CD-ROMs, Web pages and even critical software can be stored in a way that an attacker cannot modify the files. The usual configuration is for the Web server to have a read-only

port to the external hard disk while another well-protected computer has a read-write port so that the Web pages can be updated. Note that an attacker who penetrates a protected Web server can still copy data, change the copied data, and serve up the changed pages.

Internal Firewalls

Modern Web servers often serve as front ends to complex and possibly distributed applications. In this situation, a Web server often communicates with several other hosts, each of which contains particular data or performs particular computations. It is tempting to locate these computers inside of an organization's firewall for ease of maintenance and to protect these important computers. However, if an attacker can compromise a Web server, these back end systems may be penetrated using the Web server as a launching point. Instead, it is a good idea to separate the Web server back end systems from the rest of the organization's networks using an internal firewall. Then, penetration of the Web server and subsequently the Web server's back end systems does not provide access to the rest of the organization's networks.

Network-Based Intrusion Detection

Despite all attempts to patch a Web server and to securely configure it, vulnerabilities may still exist that are known to the outside world. Also, the Web server may be perfectly secure but an attacker may cleverly overwhelm the host's services such that it ceases to operate. In this kind of environment, it is important to know when your Web server has been compromised or shut down so that service can be quickly restored. Network-based intrusion detection systems (IDSs) monitor network traffic to determine whether a Web server is under attack or has been compromised or disabled. Modern IDSs have the ability to launch a limited response to attacks or notify systems administrators via e-mail, pagers, or messages on a security console. Typical automated responses include killing network connections and blocking sets of IP addresses.

Host-Based Intrusion Detection

Host-based IDSs reside on a Web server. Thus, they are better positioned to determine the state of the Web server than a network-based IDS. They provide the same benefits as network-based IDSs and in some circumstances can detect attacks better because they have finer grained access to the Web server's state. However, some drawbacks exist. An attacker that penetrates a Web server can disable a host-based IDS, thereby preventing it from issuing a warning. In addition, remote denial-of-service (DOS) attacks often disable host-based IDSs while disabling the Web server. Remote DOS attacks enable an attacker to remotely shut down a Web server without actually penetrating it. Thus, host-based IDSs are useful but they should be used in conjunction with the typically more secure network-based IDSs.

Limitations of Existing Solutions and Gaining Additional Assurance

Considerable research addresses issues of proving software secure. In some cases, it is possible to do this but it is very costly and time-consuming. Usually, by the time software is proven secure, it is obsolete and replaced with an unproven new version. Therefore, today's software is not proven secure and application of standard Web security techniques cannot guarantee that a Web server will be impenetrable.

However, a Web server can be made quite resistant to attacks by using the stated Web server security techniques in addition to using trustworthy software. By trustworthy, we mean software that can be demonstrated by some measure to be secure. The security afforded by software can be assessed by studying past vulnerabilities, using software specifically created with security as the principle goal, and using software evaluated by trusted third parties.

First, some level of assurance in software can be gained by looking at the past vulnerabilities discovered in different Web server software. The number of past vulnerabilities is an indicator of future vulnerabilities and also reflects how well the software was crafted. Trustworthiness is directly related to the quality of the software product. A poorly crafted product built explicitly to meet security needs remains a poorly crafted product and therefore not trustworthy.

Second, some companies specialize in creating very secure Web server software and some boast that no vulnerabilities have ever been discovered. Users have to balance vendor's security claims against any security-performance tradeoffs that have been made.

A third way to gain a level of assurance in software is to use evaluated and validated software. Many private-sector organizations perform third-party evaluations of commercial products in order to verify a particular level of security. One of the largest of these efforts is the National Information Assurance Partnership (NIAP). A joint venture between NIST and NSA, NIAP has helped create an international standard (ISO/IEC 15408) for specifying security requirements of IT products and evaluating them to that specification. It provides a framework by which commercial companies can have product claims tested by a third party and (if desired) obtain a certificate of validation from NIAP. Various security-enhanced products are currently under evaluation, including the firewalls of three major U.S. vendors. Look in the future for NIAP-evaluated Web server software.

For More Information

This bulletin focused on how to protect Web servers. For an understanding of how hackers break into computers and how to defend networks against them, see the May *ITL Bulletin* entitled "Computer Attacks: What They Are and How to Defend Against Them," available at http://www.nist.gov/itl/lab/bulletns/cslbull1.htm.

For detailed information on computer attacks, see the papers "Understanding the Global Attack Toolkit Using a Database of Dependent Classifiers" and "Understanding the World of Your Enemy with I-CAT (Internet Categorization of Attacks Toolkit)" located at the URL: http://www.itl.nist.gov/div893/staff/mell/pmhome.html.

General Computer Security Information:

NIST Computer Security Clearinghouse: www.csrc.nist.gov
Federal Computer Incident Response Capability: www.fedcirc.gov
National Information Assurance Partnership (NIAP): www.niap.nist.gov

Center for Education and Research in Information Assurance and Security: www.cerias.purdue.edu
Carnegie Mellon Emergency Response Team: www.cert.org